HUMBOLDT-UNIVERSITÄT ZU BERLIN

Institut für Informatik - Wissensmanagement in der Bioinformatik

Exposé zur Studienarbeit

Focused Crawling

zum Sammeln von Webdokumenten zu den Themen Molekularbiologie und Erdbeben

Betreuer: Prof. Dr. Ulf Leser, Astrid Rheinländer
Autor: Moritz Brettschneider

1 Einleitung

Das Internet ist heute die größte frei zugängliche Informationsquelle. Nahezu zu jedem Thema lassen sich eine Fülle von Informationen finden. Für bestimmte textanalytische Untersuchungen ist es notwendig große Korpora von Dokumenten eines bestimmten Themas zu sammeln. Diese sind in der Regel nicht für beliebige Themen verfügbar.

Abhilfe dafür kann das Focused Crawling schaffen. Dabei soll das Internet gezielt durchwandert werden und nur Webseiten herunter geladen, die einem vordefinierten Thema entsprechen. Anschließend sollen große thematische Dokumentensammlungen aus dem Web zur Verfügung stehen.

1.1 Aufbau des Webs

Das Internet ist wie ein gerichteter Graph aufgebaut. Dabei entsprechen die Seiten den Knoten und die ausgehenden Links den gerichteten Kanten. Ein Crawler kann das Internet, durch Besuchen der Webseiten und Folgen der in ihnen enthaltenen Links, in großen Teilen "durchwandern". Außerdem kann das Internet als ein soziales Netzwerk betrachtet werden. Ein Autor verlinkt eine andere Seite, wenn er sie für ein bestimmtes Thema als relevant einschätzt[No04]. Webseiten, welche viel von Seiten eines bestimmten Themas verlinkt werden, können in der Regel als bedeutend für dieses Thema eingeschätzt werden.

1.2 Grundprinzip eines Focused Crawlers

Einem Focused Crawler wird das Thema übergeben, welches dem des zu generierenden Textkorpus entspricht. Mit dieser Vorgabe wird von einigen ebenfalls im Voraus gegebenen Webseiten das Internet durchsucht. Dabei sollen möglichst nur Webseiten besucht werden, die dem vordefinierten Thema entsprechen. Links, welche von nicht relevanten Webseiten verlinken, werden nicht weiter verfolgt.

Das Thema muss dem Crawler in geeigneter Weise übergeben werden. Beispielsweise könnte eine Menge von Dokumenten zu diesem eine geeignete Grundlage liefern. Des weiteren ist eine Menge von Ausgangswebseiten (Seeds) notwendig um mit dem Crawlen zu beginnen. Die Seeds entsprechen den Knoten, welche zuerst besucht werden. Eine hierfür geeignete Menge erhält man zum Beispiel durch Benutzung herkömmlicher Suchmaschinen mit spezifischen Schlüsselwörtern. Die Verweise auf diese Webseiten werden in eine Schlange, dem Frontier eingefügt. Wird der Crawler gestartet, so beginnt er die gegebenen Ausgangswebseiten zu besuchen, den Inhalt und ausgehende Links zu extrahieren und dabei den Inhalt zu klassifizieren. Von Webseiten, welche als relevant klassifiziert werden, werden die Links zusammen mit dem Ankertext extrahiert und in das Frontier eingefügt. Dieser Prozess wird wiederholt bis das Frontier leer ist, es eine Unterbrechung durch den Benutzer gibt oder ein anderes Abbruchkriterium erfüllt wird.

2 Ziel

Ziel dieser Studienarbeit ist es einen Focused Crawlers zu entwickeln, welcher für die Themen Erdbeben und Molekularbiologie eine große Menge an Dokumenten liefert.

3 Vorgehensweise

3.1 Vorbereitungen

Im Mittelpunkt aller textanalytischen Arbeiten steht die logische Sicht auf ein Dokument. Diese Sicht bestimmt, was wir für ein Datenmodell nutzen werden. Davon hängt zum Beispiel ab, wie wir die Ähnlichkeit zwischen zwei Dokumenten definieren. Für unseren Focused Crawler werden nur Webdokumente in Form von HTML in englischer Sprache betrachten. Dazu prüfen wir ob mindestens 10% der Wörter die in einem Dokument vorkommen, einem der zehn häufigsten der englischen Sprache entsprechen[1]. Es wird der Text der gefundenen Webseiten extrahiert[2] und in das Vector Space Model mit der Termfrequenz übersetzt. Dazu werden wir Stoppwörter, also Worte die ein hohes Vorkommen in typischen Texten einer Sprache haben entfernen. Dazu nutzen wir eine frei zugängliche Liste der 100 häufigsten Wörter in den Büchern vom Project Gutenberg[Br10][3].Bei der Suche nach Texten über Erdbeben werden wir nicht zwischen Groß- und Kleinschreibung unterscheiden. Wenn wir jedoch nach Texten mit biologischem Inhalt suchen, nehmen Abkürzungen eine bedeutende Rolle ein. Hier werden wir nur Wörter in Kleinbuchstaben überführen, welche ausschließlich mit einem Großbuchstaben beginnen. Stemming ist das Reduzieren von Wörtern auf ihren Stamm. In [GaBo02] wurde festgestellt, dass das Stemming nur sehr geringen Einfluss auf die Genauigkeit von Support Vector Ma-

[1] Das Zipfsche Gesetz zeigt, dass bereits wenige Worte einer Sprache den Großteil an Wortvorkommen ausmachen[Ki85]. So machen im Project Gutenberg in 42'144 Büchern die häufigsten 10 Wörter bereits 24% der Wortvorkommen aus [SDA10],[Br10], Anhang 1.

[2] Dazu werden die Java-Bibliothek Boilerpipe nutzen (http://code.google.com/p/boilerpipe/).

[3] Diese sind im Anhang 1 aufgelistet.

chines (SVM) als Textklassifikatoren hatte, auch wenn komplexe Regeln für diesen Prozess angewendet wurden. Aus diesem Grund werden wir kein Stemming anwenden. Die Lemmatisierung von Wörtern ist die Zuordnung der zugehörigen grammatikalischen Grundform. Nach [LeKi02] ist die Lemmatisierung eines der langsamsten Prozesse in der Textklassifikation und bringt darüber hinaus keine substantiellen Verbesserungen in der Precision oder im Recall bei der Benutzung von SVM. Lemmatisierung werden wir also ebenfalls nicht anwenden.

Bevor wir mit dem Crawlen starten, benötigen wir eine Menge von Ausgangswebseiten, den sogenannten Seeds. Diese können auf verschiedene Weisen gegeben werden. Zum ersten können wir direkt eine Menge von URLs übergeben, welche als Ausgangspunkt für einen Crawl genutzt werden. Eine weitere Möglichkeit ist es Stichwörter anzugeben, welche man zur Suche mit Google nutzt um Seeds zu gewinnen. Zuletzt soll es möglich sein aus gegebenen Dokumenten geeignete Stichwörter zu extrahieren, mit denen man über eine Googlesuche die Seeds generiert. Die Wörter, die mit einer hohen Frequenz in den gegebenen Texten erscheinen, aber selten in herkömmlichen Texten auftreten, können dafür genutzt werden. Dafür werden wir die Wörter mit hohen TF*IDF Werten nutzen, wobei wir dazu Texte benötigen, welche nicht dem Zielthema entsprechen.

Eine weitere Vorbereitung ist nötig um den Klassifikator zu trainieren. Als Grundlage dafür benötigen wir die Texte, URLs oder die Stichwörter, welche das Thema des Focused Crawling festlegen werden. Wir werden drei Varianten eines Klassifikators implementieren. Als Baseline werden wir ein einfaches Stringmatching benutzen. Dazu benötigen wir eine Menge von Stichwörtern. Haben wir URLs oder Texte gegeben, so müssen wir aus diesen Stichwörter generieren. Das werden wir wie oben gestalten. Haben wir die Stichwörter gegeben, so müssen für eine positive Klassifikation wahlweise alle Stichwörter oder eine bestimmte Menge in einem gegebenen Dokument enthalten sein. Andernfalls ergibt die Klassifikation ein negatives Ergebnis. Außerdem werden wir Naive Bayes und eine Support Vector Machine als Klassifikator nutzen und die Ergebnisse vergleichen. Eine Implementation einer SVM steht uns mit libsvm frei zur Verfügung[ChLi]. Für diese beiden Methoden benötigen wir Dokumente, die das Thema definieren werden. Falls wir nur Stichworte gegeben haben, können wir wie oben vorgehen und über eine Google-Suche an eine gewünschte Menge von Dokumenten gelangen. Es ist ebenfalls möglich externes Wissen heranzuziehen wie beispielsweise die entsprechenden Artikel in der Wikipedia.

3.2 Focused Crawl

Haben wir das Datenmodell, die Seeds und unseren Klassifikator, können wir mit dem Crawlen beginnen. Dazu entnehmen wir den Seeds nach und nach ausgehende Links, laden die zugehörige Seite runter, extrahieren den Inhalt und klassifizieren diesen. Ist das Ergebnis positiv, speichern wir das Dokument, extrahieren die enthaltenen Links und speichern sie in das Frontier (siehe Abb. 1). Sind die Seeds abgearbeitet, werden sie genutzt um weiter durch das Web zu wandern. Sollte man Gefahr laufen, dass die Liste der URLs leer werden sollte, könnte man Links auch speichern, wenn die verlinkende Seite nur knapp negativ klassifiziert wird. Links, welche beispielsweise mit „.doc“ oder „.pdf“ enden, werden aussortiert.

Die Reihenfolge des Besuchens der gesammelten Links hat einen großen Ein-

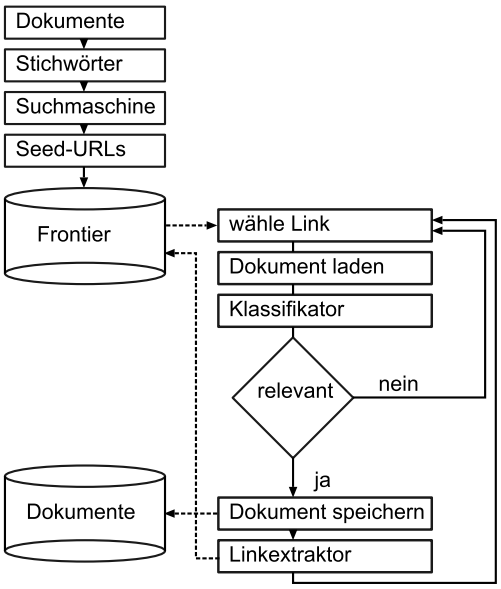

Abbildung 1: Aufbau eines Focused Crawlers

fluss auf die Geschwindigkeit mit der relevante Dokumente gefunden werden und damit bei begrenzter Zeit auch darauf welche Dokumente gefunden werden. Da wir beim Focused Crawlen nicht das gesamte Web durchsuchen wollen, können wir zum Beispiel von einer Breitensuche absehen und mit Hilfe von Algorithmen berechnen, welche Seite zunächst besucht werden soll. Ziel ist es diese Algorithmen so zu gestalten, dass möglichst viele relevante und dementsprechend wenig nicht relevante Seiten besucht werden. Beispielsweise kann der Ankertext genutzt werden um vorauszusagen wie wahrscheinlich die verlinkte Seite relevant für unser gegebenes Thema ist. So ist das Enthalten eines Stichwortes unseres Themas ein Hinweis dafür, dass es sich bei der verlinkten Seite um eine relevante handelt. Erweitern könnte man diese Schätzung, in dem man während dem Crawlen zu den in unser Datenmodell umgewandelten Ankertext das Ergebnis des Klassifikators der verlinkten Seite speichert. So könnte man während dem Crawlen einen weiteren Naive Bayes Klassifikator anlernen und zusammen mit der Klassifikation der verlinkenden Seite für die Bestimmung der als nächstes zu besuchenden URL nutzen. In [Do08] hat sich diese Kombination von Ankerklassifikator (Stringmatching) und Textklassifikation (SVM) der verlinkenden Seite als eine der besten Methoden herausgestellt. Dabei gab es für die Gesamtklassifikation die besten Ergebnisse bei einem Verhältnis von 0.2 Ankertextklassifikation zu 0.8 Textklassifikation der verlinkenden Seite. Das Nutzen von Text in der Umgebung von Links zum Vorhersagen der Relevanz verlinkter Seiten führt nicht zu Verbesserungen gegenüber dem Nutzen der Relevanz der verlinkenden Seite[MaErSt04].

3.3 Evaluation

Um uns von der Qualität des Focused Crawlers überzeugen zu können, müssen wir ihn evaluieren. Insbesondere werden wir die Crawlstrategie und die Klassi-

fikatoren analysieren und gegeneinander vergleichen. Im Information Retrieval sind die Precision und der Recall bedeutende und grundlegende Maße. Dabei ist der Recall, der Anteil der zurückgegebenen relevanten Dokumente von allen relevanten Dokumenten. Da wir jedoch nicht alle relevanten Dokumente im Web kennen können, ist dieses Maß für die Evaluation nicht zu gebrauchen. Die Precision ist der Anteil der relevanten Dokumenten an allen zurück gegebenen Dokumenten. Dieses Maß werden wir nutzen um die Klassifikatoren zu evaluieren, wobei wir uns hier auf Stichproben beschränken müssen. Es ist aufgrund der Fülle nicht möglich alle zurück gegebenen Dokumente per Hand zu überprüfen. Außerdem werden wir die Klassifikatoren auf per Hand klassifizierte Dokumente anwenden.

Zur Evaluation der Crawlstrategie werden wir die Harvestrate nutzen. Diese ist das Verhältnis von den gefundenen relevanten Dokumenten zu den insgesamt besuchten[ChBeDo99]. Je höher diese Rate ist, desto weniger Webseiten wurden umsonst besucht. Auch hier werden wir die Verschiedenen Klassifikatoren gegeneinander Vergleichen.

Den Ankertextklassifikator können wir bei jeder Klassifikation der verlinkten Seite evaluieren, wobei in der Auswertung berücksichtigt werden muss, dass sich die Klassifikation wärend des Crawlens verändert und sich die Precision des Naive Bayes Klassifikator ständig verbessern sollte. Hier können wir das Ziehen der Links in der Reihenfolge wie sie extrahiert werden mit dem Sortieren nach der Relevanz der verlinkenden Seite und unserem Ankertextklassifikator vergleichen. Ebenfalls werden wir den Naive Bayes Klassifikator mit dem Stringmatching vergleichen.

A Anhang

A.1 häufigste Wörter der englischen Sprache im Project Gutenberg

In [Br10] sind die 100'000 häufigsten Wörter aus 42'144 Büchern aus dem Project Gutenberg im Jahr 2010, mit der Anzahl ihrer Vorkommen aufgelistet. Insgesamt wurden 2'880'579'249 Wörter gezählt[SDA10].

Rang	Wort	Anz. der Vorkommen
1	the	183'605'611
2	of	101'290'462
3	and	91'277'015
4	to	76'356'196
5	a	59'953'823
6	in	51'838'219
7	that	33'090'620
8	i	32'670'756
9	he	29'867'935
10	it	28'965'492
	Summe	688'916'129
	Alle Wörter	2'880'579'249

Abbildung 2: Die zehn häufigsten Wörter im Project Gutenberg[Br10].

Die 10 häufigsten Wörter in diesen Büchern machen also rund 24% (688'916'129 / 2'880'579'249 = 0.239158888) der Wortvorkommen aus.

Die 100 häufigsten Wörter sind:
the, of, and, to, a, in, that, i, he, it, was, his, with, is, for, as,you, had, not, be, on, at, by, but, s, her, this, which, or, from, have, she, they, all, him, are, were, we, one, my, so, an, their, there, no, me, if, who, when, said, them, been, would, will, what, out, up, more, do, any, t, then, into, has, some, man, other, your, now, our, could, its, time, very, than, about, may, can, upon, only, like, little, these, see, gutenberg, two, such, great, project, made, did, well, before, after, should, work, must, us, over, good

Literatur

[AlHa08] Jesse Alpert, Nissan Hajaj; *We knew the web was big...*, http://googleblog.blogspot.com/2008/07/we-knew-web-was-big.html, Stand: 7.7.2011, 2008.

[ChLi] Chih-Chung Chang, Chih-Jen Lin; *LIBSVM – A Library for Support Vector Machines*, http://www.csie.ntu.edu.tw/~cjlin/libsvm/ Stand: 18.7.2011.

[Do08] Ignacio García Dorado; Masterarbeit: *Focused Crawling: algorithm survey and new approaches with a manual analysis*, Lund University, 2008.

[GaBo02] Tanja Gaustad, Gosse Bouma; Accurate Stemming of Dutch for Text Classification, In: *Computational Linguistics in the Netherlands 2001*, 104–117, 2002.

[LeKi02] Edda Leopold, Jörg Kindermann; Text Categorization with Support Vector Machines. How to Represent Texts in Input Space?, In: *Machine Learning*, 46, 423-444, 2002.

[MaErSt04] Benjamin Markines, Fulya Erdinc, Lubomira Stoilova; Seminarbericht: *Focused Crawlers vs Accelerated Focused Crawlers*, Indiana University, 2004.

[No04] Blaž Novak; A survey of focused web crawling algorithms, In: *SIKDD*, Multiconference IS, Ljubljana, 2004.

[SDA10] *Semantic Depth Analyzer on 1o1.in*, http://1.1o1.in/en/webtools/semantic-depth, Stand 22.08.2011, 2010.

[Br10] Luc Brunet; *Frequential Dictionary of English language*, http://1.1o1.in/semanticdepth/freq100000.pdf, Stand 22.08.2011, 2010.

[Ki85] Geoff Kirby, Zipf's Law, In: *UK Journal of Naval Science Volume 10*, Nr. 3, 180-185, 1985.

[ChBeDo99] Soumen Chakrabarti, Martin van den Berg, Byron Dom; Focused crawling: a new approach to topic-specific Web resource discovery, In: *Proceedings of the 8th World-Wide Web Conference*, 1999.